

INTERLINKING RELATED DIVERSE MEDIA IN A DIGITAL  
LIBRARY

A Thesis

by

MANAS SOURAVA SINGH

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2006

Major Subject: Computer Science

INTERLINKING RELATED DIVERSE MEDIA IN A DIGITAL  
LIBRARY

A Thesis

by

MANAS SOURAVA SINGH

Submitted to the Office of Graduate Studies of  
Texas A&M University  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Richard Furuta
Committee Members,	Frank Shipman
	Eduardo Urbina
Head of Department,	Valerie E. Taylor

May 2006

Major Subject: Computer Science

## ABSTRACT

Interlinking Related Diverse Media in a Digital Library. (May 2006)

Manas Sourava Singh, B.E., R.E.C Rourkela, India

Chair of Advisory Committee: Dr. Richard Furuta

Digital libraries are widely used for organizing and presenting large collections of artifacts. However, as the digital libraries grow in size, it is becoming increasingly difficult for the user to find all the resources related to his topic of interest. It is labor intensive, time consuming and error prone to identify and link related materials manually. Thus it is important to develop automatic techniques to help the user discover and view the related resources that are available in the digital library.

We have implemented an automatic interlinking mechanism for a music digital library system that spans across batch, online and on-demand phases. Since the task of generating the related links is resource and time intensive, distributing the whole process across these three phases significantly reduces the runtime overhead and improves the response time. This mechanism allows the system to display very large texts, with keywords identified and hyperlinked, with no perceivable delay to the user. Storing the artifacts in a structured manner and using the structural metadata to generate interlinkages allows us to create these links across diverse media like images, audio files, music scores, texts, etc. The implemented interlinking technique also scales well with a rapidly changing collection. The related links are displayed on demand, using AJAX technology. This allows the user to view these links without leaving the text, thus ensuring minimum disruption and continuity of action. We also have developed a generic interlinking framework which abstracts the domain independent logic for generating and displaying related links. This generic interlinking framework can be used by domain specific digital libraries to support interlinking of related resources.

## DEDICATION

To my dear Mother and Father  
For their unconditional love.

## ACKNOWLEDGEMENTS

I would like to thank Dr. Richard Furuta for his guidance, support and encouragement in completing the thesis. This thesis would not have been possible without his insightful comments and guidance.

I would also like to thank my committee members Dr. Eduardo Urbina and Dr. Frank Shipman for their support and faith in my work. I am grateful to Dr. Urbina for devoting ample time for this project despite his busy schedule and answering my numerous doubts and questions.

I thank Ananth Kini and Neal Audenaert, from the Department of Computer Science, for sharing their knowledge and time towards discussions surrounding the thesis.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	iii
DEDICATION .....	iv
ACKNOWLEDGEMENTS .....	v
TABLE OF CONTENTS .....	vi
LIST OF FIGURES .....	viii
1. INTRODUCTION .....	1
1.1. Objective .....	1
1.2. Motivation .....	2
1.3. Organization of the Thesis .....	3
2. BACKGROUND .....	4
3. RELATED WORKS .....	7
3.1. Digital Library .....	7
3.1.1. Music Digital Libraries .....	7
3.1.2. Open Source Digital Library Systems .....	8
3.2. Automatic Interlinking of Related Information .....	8
3.2.1. Linking Using Authority List .....	8
3.2.2. Linking Using Document Structure .....	10
3.2.3. Linking Using Information Retrieval Techniques .....	10
3.2.4. Simultaneous Searching and Reference Linking .....	11
3.2.5. Integration Using Virtual Collection .....	12
4. APPROACH .....	13
5. USAGE SCENARIO .....	17
6. SYSTEM DESIGN .....	25
6.1. Technology Components .....	25
6.2. System Objectives .....	26
6.3. Organization of the Digital Library .....	27
6.4. Implementation of the Digital Library .....	28
6.5. Implementation of Interlinking .....	33
6.5.1. Maintaining the Keyword List .....	34
6.5.2. Document Keyword Mapping Batch Job .....	36

	Page
6.5.3. Runtime Display of Texts with Dummy Hyperlinks .....	38
6.5.4. Display of Related Links Tooltip.....	39
7. DISCUSSION AND FUTURE WORK.....	43
8. CONCLUSION.....	46
REFERENCES .....	47
VITA.....	51

## LIST OF FIGURES

FIGURE	Page
1 System displaying list of instruments .....	17
2 List of images for instrument Arpa.....	18
3 Text with keywords identified and hyperlinked .....	19
4 Related links for instrument Guitarra and a sample audio file. ....	20
5 Related links for song <i>Mira Nero De Tarpeya</i> displayed using a tooltip. ....	21
6 Music score for the song <i>Mira Nero de Tarpeya</i> .....	22
7 Related links for the composer Mateo Flecha.....	23
8 Related links for the song links the composer to the songs that he has composed .....	24
9 User interface for uploading artifacts .....	29
10 Model view controller architecture .....	30
11 Data model for the metadata repository .....	31
12 The keyword list data model.....	34
13 Schematic diagram for maintaining keyword list and synonym list.....	35
14 Generation of document keyword map.....	37
15 A sample keyword document mapping table.....	38
16 Display of texts with dummy hyperlinks .....	39
17 A sample AJAX response .....	41
18 Generation and display of related links .....	42



# 1. INTRODUCTION

## 1.1. Objective

Digital libraries are widely used for organizing and presenting large collections of artifacts. However as the digital libraries grow bigger in size, it is becoming increasingly difficult for the user to find all the resources related to his topic of interest. Thus it is important to develop mechanisms to help the user discover and view the related resources that are available in the digital library.

This thesis proposes to build a music digital library system that will automatically provide interlinking and navigation between related but diverse media. It will support addition of diverse media types like text excerpts, audio files, images, playable scores, etc., through suitable Web interfaces. It can be used to store and browse information pertaining to instruments, songs, dances, and composers that either Cervantes refers in his text or are inspired by Cervantes' work [27]. It will store these artifacts in a structured and organized manner and will create browsing indexes to facilitate information discovery and navigation.

An important objective of this thesis is to facilitate automatic interlinking of text and related information. It should be able to maintain and identify the keywords that represent instruments, songs, dances, composers, etc., in texts and link them to the related materials present in the digital library. This related information can be of diverse media types like text, image, audio, playable scores, etc. It should also provide easy navigation to the related artifacts. The library should be able to maintain and create these interlinkages while allowing for progressive addition of new information. For example, it should be able to identify the name of the instruments for which information exists in the digital library, and automatically link these words or phrases to the related materials that provide the authoritative source of information for the instrument.

## 1.2. Motivation

The digital library will provide resources for scholars investigating the impact of music on Cervantes and the influence of his work on subsequent musicians and authors. It aims to provide a complete picture by making digitally available resources spanning across categories like instruments, songs, dances, composers and bibliographical records. It will provide access to text excerpts, images, audio files, playable scores, bibliographical information, etc., for items under each category.

The digital library will provide automatic interlinking, which will help the users in discovering and viewing the related information present. Without these links it would be very difficult for the user to know about the existence of the related information. Thus interlinking provides a comprehensive view of the library and enhances its value for the users. Besides it would be very labor intensive and time consuming to identify and link related materials manually. Humans may be inconsistent in the selection of links, which may undermine “coherency” of the interlinking [19].

One of the key strength of digital libraries is their ability to support varied interpretive perspectives of scholars [10]. A digital library can be used to present the resources through different perspectives to cater to the varied interests of the scholars. Some scholars are interested in Cervantes’ texts and the influence of music on his writings. The digital library will enable them to browse through excerpts of Cervantes’ texts. It will highlight the terms referring to instruments, songs, dances, composers, etc., for which information exists in the digital library and provide links to the related artifacts. These scholars can browse through the texts, identify the references to the musical terms and navigate to the authoritative sources for these musical terms. This helps them to view the musical artifacts in proper perspective. On the other hand some scholars are more interested in music and how it has influenced Cervantes. The digital library will enable them to browse through different musical artifacts like instrument, song, dance, composer, etc. It will provide them navigational links from these musical artifacts to the

relevant Cervantes text excerpts enabling them to traverse from music to the related texts. This gives them a better understanding of the impact of music on Cervantes work.

The ability to maintain and identify the occurrences of the keywords and to provide link to the related media can be used to integrate different digital libraries. The digital library can import a list of keywords and their authoritative source from another library. Using this information, it can link the occurrences of these keywords to the corresponding digital library. This can help in integrating related but different digital libraries, which will help to realize the true potential of digital libraries.

### **1.3. Organization of the Thesis**

This thesis will arrange the ensuing discussion under six main sections. Section 2 presents background material on Cervantes project that helps to place the thesis discussion in proper context. Section 3 discusses the related research on automatic hyper-linking and interlinking of related information. Section 4 gives a very high level description of the proposed interlinking mechanism. Section 5 presents a usage scenario from the user's perspective. Section 6 gives a detailed system level overview of interlinking mechanism. Section 7 presents a discussion on the working of the interlinking technique and proposes future work and applications. Section 8 concludes the thesis.

## 2. BACKGROUND

The Cervantes Project has its objective in providing comprehensive on-line resources on the life and works of the author Miguel de Cervantes Saavedra (1547 – 1616) [34]. He is best known for authoring *Don Quixote*, which is considered as the first modern novel and was first published in two parts in 1605 and 1615. Cervantes, being a central figure in Hispanic Literature, attracts large research interest. Although individual researchers focus on very detailed and specific research questions, these questions collectively span across the whole gambit of Cervantes' life and its impact on society. For example some researchers are interested in the works authored by Cervantes and commentary on those works, while others focus more on historical and biographical records and some others on influence of music on Cervantes. To support such diverse research, the Cervantes project aims to make digitally available the entire range of resources ranging from Cervantes' works and commentary on those works, historical documents of his time, bibliographical records and information about his life, information about the contemporary culture and its influence on Cervantes and the scholarly and popular artifacts that are based on or inspired by his works [4][5][15]. With this purpose in mind, the Cervantes project is developing a suite of tools that can be categorized under six sub-projects: bibliographic information, textual analysis, historical research, music, *ex libris* and textual iconography. The proposed thesis falls into the music category and will help the scholars add and browse information pertaining to the confluence of music and Cervantes. The following is a brief description of each of the sub-projects, which provide the background and a high level context for the music digital library system.

The Cervantes project maintains and publishes a comprehensive bibliography of scholarly publications based on Cervantes' life and work [33]. These comprehensive bibliographical records provide an insight into the impact and influence of Cervantes on subsequent authors and musicians. In order to maintain and analyze such a huge collection of bibliographical records, the project has developed a flexible database-driven tool supporting annotations, taxonomies, and multiple editors.

In order to cater to scholars interested in Cervantes' work, the Cervantes Project maintains various editions of *Don Quixote*. To make these editions digitally available and to support textual analysis, the Cervantes project has developed an electronic variorum edition (EVE), a reader's interface (VERI) and a variant editor (MVED) [15]. The EVE is an electronic edition that will consist of the significant early editions of *Don Quixote*. The MVED assists scholars in detecting, analyzing and editing the variances among various versions of *Don Quixote*. The VERI enables users to access the texts along with the associated annotations and emendations. The Cervantes project has also used the Interactive Timeline Viewer to visualize the variants [24].

The Cervantes project is developing tools to make digitally accessible 1600 official documents pertaining to the life of Cervantes and his family [4][32]. The tool will support identification of named entities like people, places and numerical entities like dates, etc. This will give a better understanding of the structure of the historical documents and will aid in the development of collection visualization tools. This will also provide scholars, with a better understanding of the historical context of Cervantes' works. This will help them to view the works in proper perspective and appreciate the inspiration and motivation behind it.

The Cervantes project is building a comprehensive collection of illustrations from various editions of *Don Quixote*. It has digitized more than 4000 images from the 74 illustrated editions of *Don Quixote*. To preserve and analyze their artistic features and literary context these illustrations are encoded with detailed metadata information. The project is developing number of collation strategies to help analyze these images from different perspectives. Some of the collation methods being developed are "book-based collations that allow the illustrations to be placed in their original physical, narrative or thematic context, natural collations that group illustrations by author, style, size, etc., and custom collations created or tailored by the users" [4]. This helps scholars in analyzing the artist's interpretation of *Don Quixote* over a span of time and the cultural, political and ethical factors that have influenced these interpretations. Another project,

using similar technology, is making digitally accessible more than 1300 *ex libris* (bookplates) inspired by or based on *Don Quixote*.

The influence of music on Cervantes work and its subsequent interpretation and impact on musicians is another area of research for the scholars. Information has been collected on various musical instruments cited by Cervantes and accomplishment of the scores related to his work [27]. This includes detailed explanation about the musical instruments including image files, audio files, and extracts of Cervantes' text having reference to these instruments. The collection also includes digital editions of scores related to Cervantes' text, including sound files, playable scores, text excerpts, bibliographical information about composers and different articles explaining the significance of the relationship between music and poetry in Cervantes' works. Besides the impact of music on Cervantes, the collection also intends to provide a complete catalogue of musical compositions based on Cervantes' texts. The work inspired by Cervantes will be categorized by genres, countries and musical period. This will enable scholars to study the influence and interpretation of Cervantes work by the musicians. The focus of the thesis is to develop a system that will allow scholars to easily add information related to Cervantes and music and to make this information available online through a well structured browsing index.

### 3. RELATED WORK

The related work is broadly grouped into two categories. The first describes work related to the digital library, specifically various music collections and open source infrastructures for creating digital libraries. The second gives an overview of various approaches related to automatic hyper-linking and interlinking of related information.

#### 3.1. Digital Library

This section gives a brief overview of works related to music digital libraries and open source digital library systems.

##### 3.1.1. *Music Digital Libraries*

Growth in access to the Internet has stimulated the growth of digital libraries that organize, store and provide means to access information stored in digital format. Generally the digital libraries are highly focused and cater to niche areas. Consequently libraries focused on music have been developed to cater to the requirements of music scholars, musicians, and general public interested in music. The Indiana University Digital Music Library provides access to music in different formats including sound recordings, musical scores, and computer score notation files catering to a wide variety of audiences including computer scientists, musicians and research scholars [13]. Bainbridge, et al., have developed a comprehensive suite of tools that support collecting different representations of music like facsimile images of scores, MIDI files, audio files, etc., inter-conversion between these representations and searching based on combined musical and textual criteria [7]. McNab, et al., have developed the Meldex system, which uses a few sung or hummed notes to search and retrieve matching songs present in the library [22] [6]. Cunningham, et al., focus on “music information needs or information behavior” of a target user group, i.e., people searching for popular music [11]. Their focus is to better understand the requirements of common man, by studying

their browsing, searching, and usage behavior, so as to design user friendly digital libraries.

### *3.1.2. Open Source Digital Library Systems*

With the popularity of digital libraries various open source systems have been developed that provide ready to use software infrastructure for creating digital libraries. Greenstone is one such open source digital library software system [35][36]. It provides comprehensive set of tools for construction and presentation of information collections. Greenstone automatically creates full text indexes from the document text and from metadata elements such as title and author. It builds browsing indexes based on the metadata. It uses Unicode, allowing any language to be processed and displayed in a consistent manner. Another popular open source digital library system is DSpace [12] [38]. DSpace is a combined venture by MIT and Hewlett Packard and targets digital content needs of institutions by providing “detailed workflow for digital works submission, means to distribute institution’s digital works over the Web through a search and retrieval system and to preserve digital works over the long term” [38].

## **3.2. Automatic Interlinking of Related Information**

Interlinking related information provides a comprehensive view of the available resources related to a topic. This helps the user to realize the true potential of the digital library and hence enhances its overall value. The following subsections briefly discuss the various approaches for creating automatic hyper-linking and interlinking of related information.

### *3.2.1. Linking Using Authority List*

In this approach one or multiple authority lists are maintained consisting of the terms that need to be hyperlinked and the corresponding authoritative sources. These authoritative sources contain detailed information about the terms in the authority list.



The Perseus Project's London collection uses multiple authority lists to interlink related information that gives context to the words and help the users develop their own interpretation and mental model [10]. The main motivation for creating interlinkages is to make it easier for twenty-first century readers to understand and analyze the London Collection. The London collection contains many articles in Latin and Greek and current English readers generally tend to lack awareness about the Greek and Latin languages. Also the collection is characterized by frequent use of people, places and topics that may be unfamiliar to general readers. The London collection links the Greek and Latin words to grammatical analysis, dictionaries and other linguistic support tools, helping a wider audience to understand and appreciate them. The London Collection also links the names of the people, places and topics to reference materials that can give more information about them. To achieve this, the London Collection has added many reference materials to the collection and has created a unified authority list. For example, it uses the *British Directory of National Biography*, which contains brief biographies for about 34,000 famous people. It uses *London Past and Present*, an encyclopedia, for information about places. It generates the links at runtime, using an efficient algorithm that matches each text against a large multiword authority list.

The SCALE – Services for a Customizable Authority Linking Environment – project is developing tools to integrate collections within the National Science Foundation's National Science Digital Library through “terminological linking” [28]. It maintains its authority list as a list of terms associated with Internet authoritative reference URIs. The reference linking process is a two step process. The first step is finding references by matching a document against the list of available reference terms. The second step involves displaying the processed document to the user by incorporating reference links to the authoritative reference directly into the document. Only the relevant reference links are displayed using statistical methods and user preferences. The SCALE project has implemented different display methods like highlighting matched terms and displaying links on mouse over, directly linking matched terms to references or displaying matched terms linked to references in a separate area of the interface.

### 3.2.2. *Linking Using Document Structure*

This approach uses lexical analysis and the structure of the document to identify the anchors, rather than depending on authority lists to identify them. Once identified, these anchors are linked to the available services which can act on them.

The Digital Library Service Integration (DLSI) project uses this approach [9]. DLSI identifies the anchors for linking in two ways. First the “wrappers” use the knowledge of the structure of the document to parse and identify the anchors. The structure of the document can be specified using templates, XML markup, or parsing rules. Second it also identifies additional anchors by lexically analyzing the document. Once the anchors are identified, links are automatically generated to available services based on the type of anchor and the specified rules. For example if the term is a proper noun it can be linked to glossaries/thesauri which can provide related additional information.

The Key Linking framework presents the available services as a list of verbs and considers the linking process as “enacting a verb with a data element” [30]. Verbs can range from the very simple “define,” which simply shows a description of the data element, to the complex “send flowers,” which acts on the data element and can be used to initiate transactions over the Internet or to lookup addresses, etc. The verbs are implemented using JavaScript and linked to suitable services. The system builds link to available services by selecting suitable verbs based on the data type of the selected term.

### 3.2.3. *Linking Using Information Retrieval Techniques*

Information retrieval techniques like the vector space model can be used to determine the statistical similarity between text passages [31] [37]. Hence they can be used to identify and create links between similar documents. This method is suitable for collections that have no evident structure and lack metadata.

The New Zealand Digital Library has developed two systems, Kniles and Phrasier, that automatically generate links between similar documents using information retrieval

techniques [19] [26]. Key phrases are identified using machine learning techniques [14]. This system supports linking between related documents as a whole. It also supports topic-based linking by using the key phrases as anchors and linking the key phrases to the related documents. For any given document a list of related documents ranked by similarity is retrieved and displayed. For each identified key phrase, it also displays a list of related documents based on their similarity to the key phrase.

#### *3.2.4. Simultaneous Searching and Reference Linking*

Some projects have used simultaneous searching and reference linking to support integration between the collections and multiple supporting information resources.

Mishco, et al., have implemented a system that can simultaneously search multiple information resources and efficiently link the search results to other supporting information resources located elsewhere [23]. The system has access to number of bibliographic databases and carries out searches asynchronously and simultaneously over all of them. It matches the search results against the CrossRef database, which is a metadata database of the available supporting information. If the journal is available then the system automatically adds a link to the full text of the journal while displaying the search results. The OpenURL format is used to create the automatic link to the full text article. A handle server is used to analyze the OpenURL and redirect the user to the specific full text information at the publisher site.

Instead of simultaneous asynchronous searching across multiple databases, Lagoze, et al., have used a common metadata repository [20]. The metadata is maintained by several methods including automatic harvesting using Open Archives Initiative Protocol, ingest via FTP, e-mail, or upload and ingest by direct entry. This common metadata repository is used by search and discovery system to retrieve related documents which may span across various collections.

### *3.2.5. Integration Using Virtual Collection*

Geisler, et al., have proposed linking related materials by creating virtual collections. Virtual collection helps to logically tie together related information irrespective of their actual physical organization [17]. The virtual collections can be organized using various parameters like resources that share a topic or some other significant attribute, or those used for a specific purpose, etc. This flexibility enables linking of information in different contexts allowing the users to view the collection from various perspectives.

## 4. APPROACH

The approach taken to interlink related information in the digital library is to maintain a list of terms that need to be interlinked. This list of terms will be created and maintained by the digital library and will consist of names of different musical elements like instrument, song, composer, dance, etc., for which information exists in the music digital library. The references to these terms are identified and linked to the related information. The library uses the metadata of the artifacts it contains to identify the authoritative sources and generates links to those sources. The use of metadata enables the digital library to link to diverse media types like text, image, audio, etc.

The steps involved in generating links to the related information are as follows:

1. Maintain a keyword list consisting of the terms that need to be linked. When a new item is added to the digital library, add a corresponding entry to the keyword list. The item may be known by different names. To be able to identify these variations and link them together it should also maintain a synonym list containing these variations in names.
2. When a new resource is added to the digital library, store the corresponding metadata in a structured manner, so that it can be used to find the related information.
3. For each term in the above mentioned keyword list, find references in all the documents present in the library. This information can be used to build a document-term index that maps each document to the list of terms it contains that need to be linked.
4. For each document, refer to the term-document index to find the terms that need to be linked.

5. For each term that needs to be linked, use the metadata repository to find the available related artifacts. The synonym list may also be used to resolve the variations in the names and link these to the related information.
6. Once the related artifacts for each term is identified, find all the references to the term in the document and add links to the related artifacts.

The above steps can be implemented entirely in batch mode or entirely at runtime. Creating links in the batch mode speeds up the response time. Since the key terms are identified and the links to the related information are generated offline, there is no additional run time cost in displaying these links to the user. However, the offline approach is less flexible as it involves modifying the actual documents. This does not scale well with rapidly changing collection as new linking must be added and some old links removed from the documents. Moreover the related links represent the state of the collection when the batch process was executed, which may not represent the current state. In contrast, the runtime approach does not involve any modification to the actual document and hence offers greater flexibility. Since the related links are generated at runtime, they represent the current state of the collection. Also it is possible to control the type and extent of interlinking based on different criteria like user preference. However, creating the links entirely at runtime involves high overhead in terms of runtime costs. This may slow down the response time and adversely affect the user experience.

Thus the approach used in this digital library is to distribute the task of creating links across both batch and runtime processes. The document term mapping is created by the batch process by matching the terms in the keyword list and the synonym list against all the documents present in the digital library. However, the actual creation of links is done at runtime. Since the related links are generated at runtime, no modification of the document is required. When a user requests a document, the document term mapping is used to find out the terms in the document which needs to be linked. The document term index obliterates the need for matching all the terms in the keyword and synonym list

against the document and hence reduces the runtime overhead and improves the response time. The only runtime overhead is to find the related artifacts for the terms identified from the metadata repository and to generate links to those artifacts. This has the advantage that the generated links represent the current structure of the digital library.

This process is further optimized by deferring the actual generation of related links until the user explicitly expresses his interest. A page may contain many terms with interlinking but the user may or may not be interested in all of them. In the worst case the user may not be interested in any of these links. In such cases the runtime overhead associated with generation of links could have been reduced or completely eliminated. The approach taken by the system is to identify the terms and provide a dummy hyperlink, i.e., a hyperlink without any target, while displaying the text to the user. When the user explicitly expresses his interest by clicking on any of those hyperlinks, a request is sent to the server for the related links using AJAX (Asynchronous Java Script and XML) [29]. The response is parsed and the related links are displayed using a tooltip just below the keyword. Using AJAX enables the system to refresh just a portion of the page, the tooltip in this case, without submitting the whole page. This ensures continuity of action for the user and minimum interruption. This approach of generating related links on demand reduces the response time and provides a better user experience.

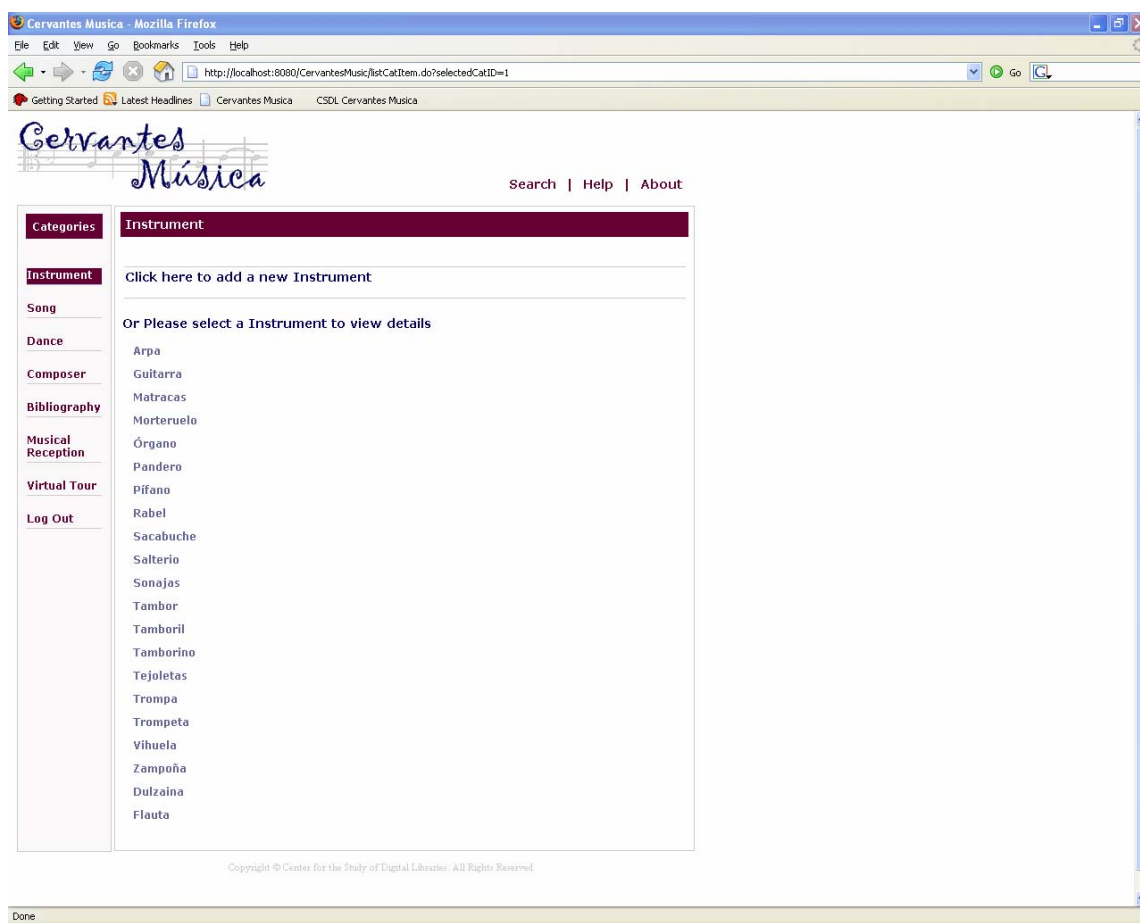
Some of the tasks in generating related links are domain specific and will differ from one digital library to another. However, many of these tasks are domain independent and will most probably be the same across digital libraries. The system abstracts the domain independent part as a generic interlinking framework that can be used by a domain specific digital library to provide the linking. This generic interlinking framework provides the basic infrastructure for creating and displaying the links. However it is the responsibility of the digital library to maintain the list of terms that need to be linked and to provide the related links for those terms; this is domain specific and hence cannot be abstracted as a part of the generic interlinking framework. However, once the digital

library is able to do this, it can use the interlinking framework to locate the references of the terms, provide dummy hyperlinks, handle user clicks using AJAX, send the response back, and display the links using the tooltip. The interlinking framework provides the basic infrastructure and displays whatever related link that is provided by the digital library. Hence the digital library is free to produce the related links in a domain specific and implementation specific manner and then to use the generic interlinking framework to actually display it to the users.



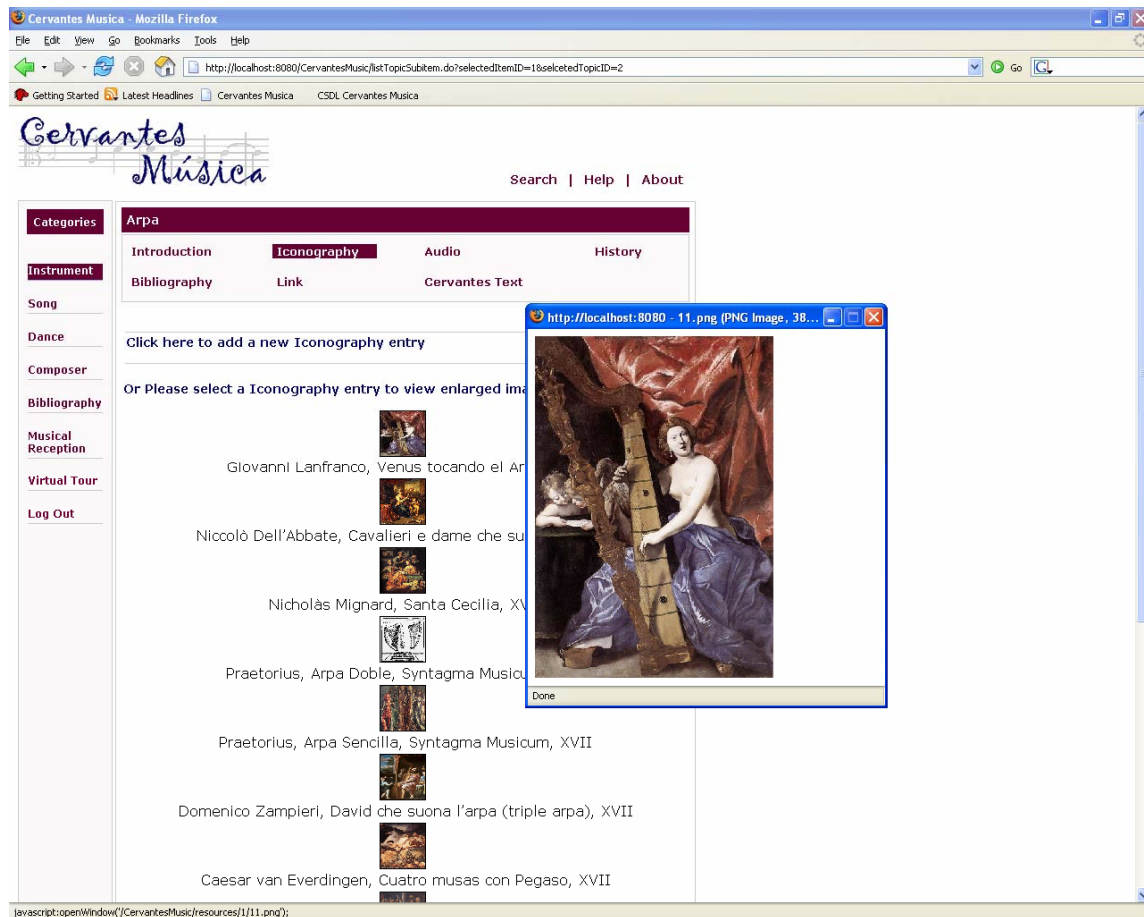
## 5. USAGE SCENARIO

This section presents a usage scenario from the user's perspective, to illustrate the working and the benefits of the interlinkages that are created automatically by the system. Suppose the user is looking for information pertaining to an instrument, Arpa. He can browse the list of instruments, for which information is available in the digital library, by clicking on the "Instruments" category on the left menu. Figure 1 shows the list of instruments.



**Figure 1: System displaying list of instruments**

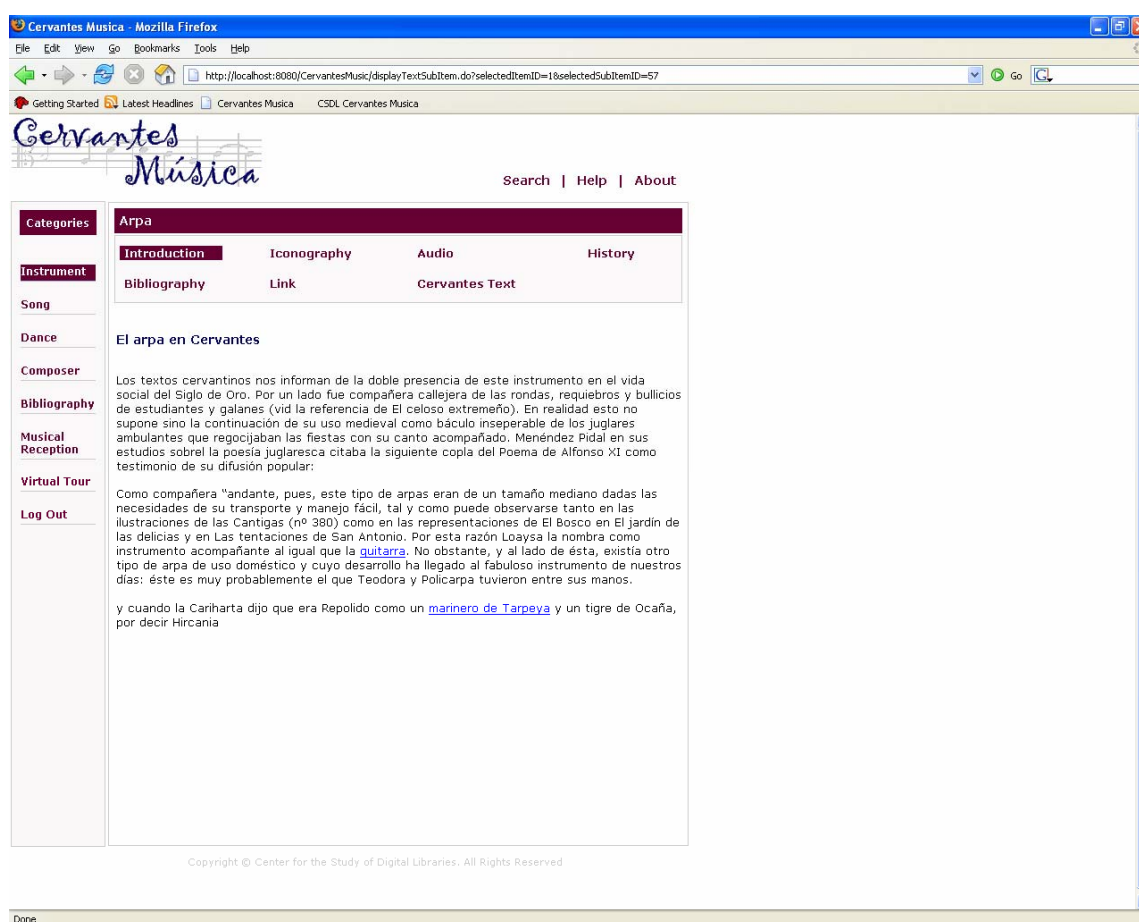
The user can access resources related the instrument Arpa by selecting “Arpa” from the list. The user has access to images, audio files, list of bibliographical records, external links, Cervantes’ text excerpts, and introductory texts related to Arpa. Figure 2 displays the list of images available for the instrument Arpa.



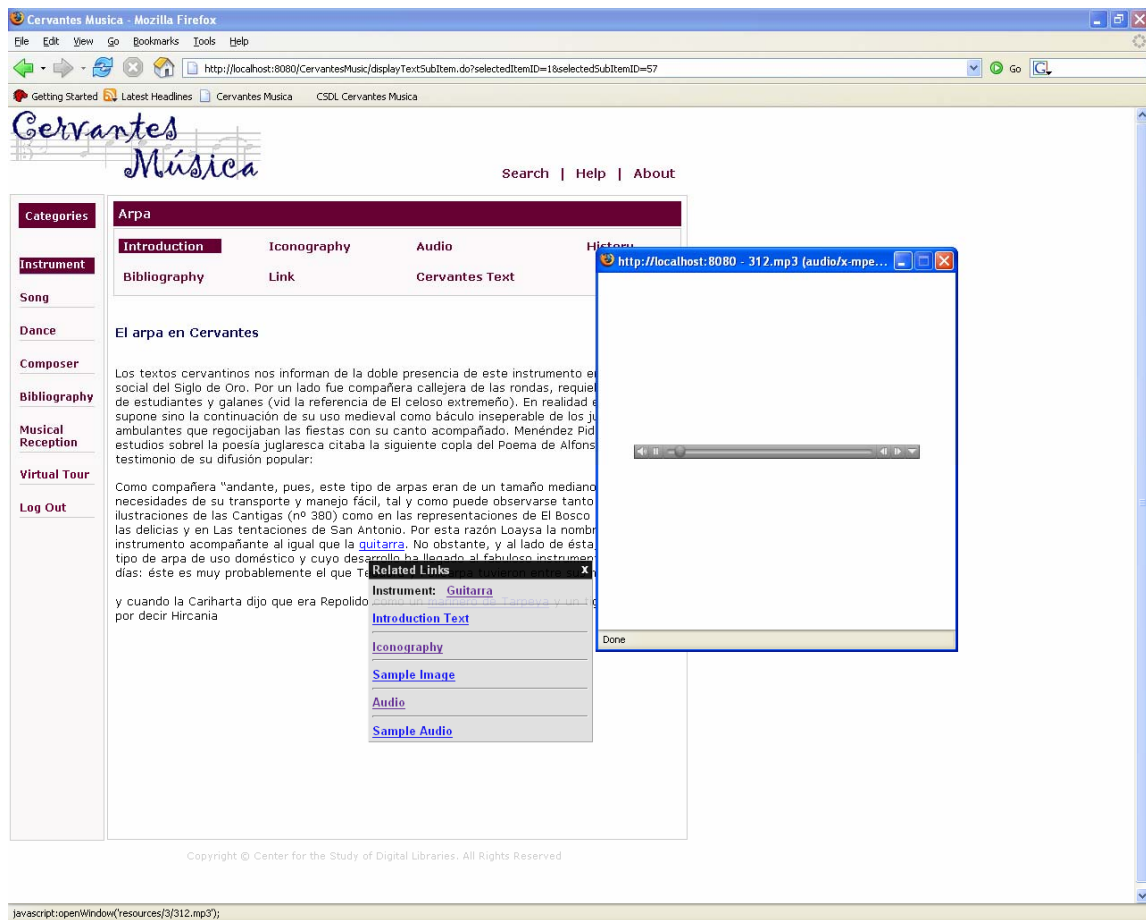
**Figure 2: List of images for instrument Arpa**

Figure 3 shows an introductory text for Arpa. The system identifies the keywords and links them to the related resources. This text has reference to another instrument, Guitarra and a song, *Mira Nero de Tarpeya*. The system identifies these two keywords and displays them in a contrasting color along with the hyperlinks. The user can access the related resources by clicking on these keywords. Figure 4 shows the related

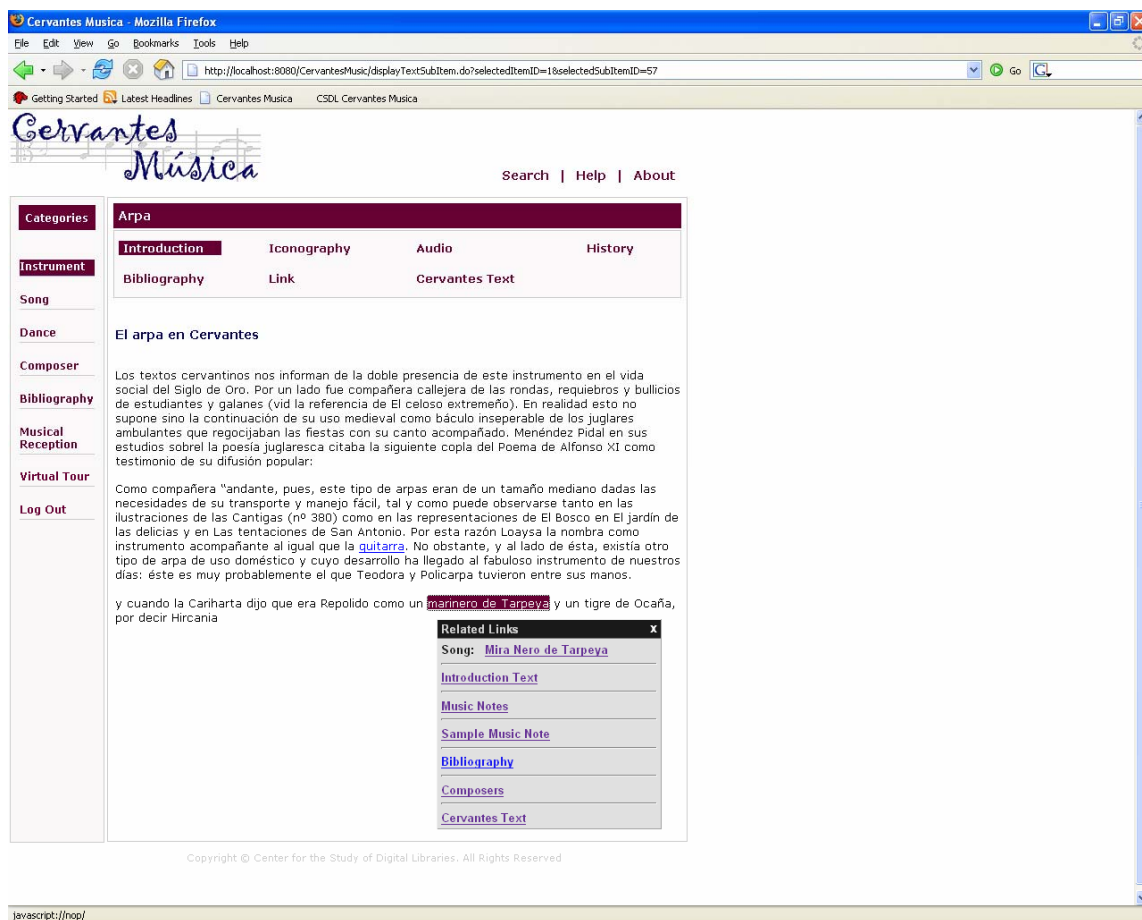
resources for the instrument Guitarra, displayed using a tool tip. The user can navigate to these related resources by clicking on these links. The user can also view sample images and hear to sample audio files without leaving the current page. The Figure 4 shows the user listening to a sample audio file for the instrument Guitarra. The user can also click on the other highlighted keyword “marinero de Tarpeya” to access related resources related to it. The system recognizes this is an alternate name for the song *Mira Nero de Tarpeya* and links it to the resources related to the song *Mira Nero de Tarpeya*, as shown in Figure 5 .



**Figure 3: Text with keywords identified and hyperlinked**



**Figure 4: Related links for instrument Guitarra and a sample audio file.**



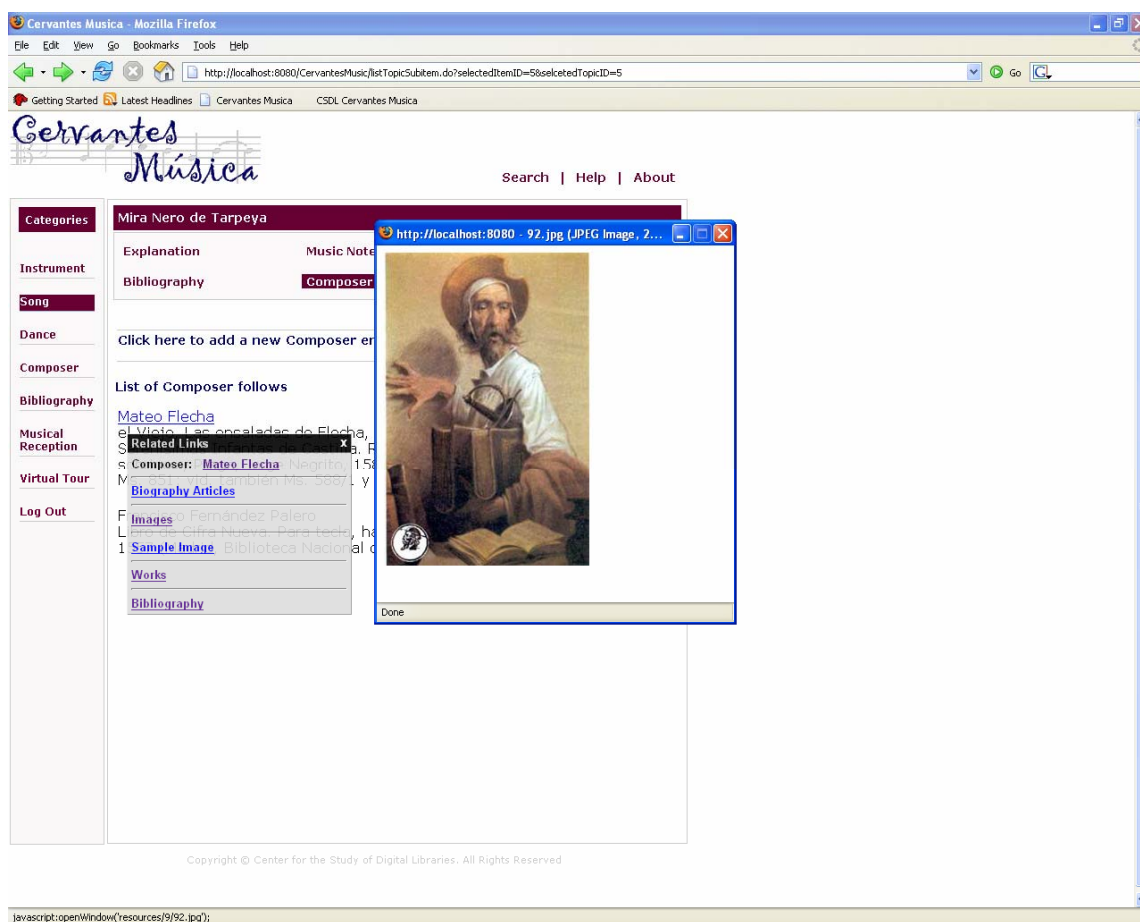
**Figure 5: Related links for song *Mira Nero De Tarpeya* displayed using a tooltip.**

The user can navigate from the instrument Arpa to the song, *Mira Nero De Tarpeya*, using these links. The user can access introductory texts, Cervantes' texts excerpts, music notes, audio files, and bibliography records pertaining to this song. The Figure 6 shows a music note for the song, *Mira Nero De Tarpeya*. The user can also view the list of composers for this song. And if the digital library has resources related to these composers, interlinks will be automatically created by the system. The Figure 7 shows the links to the resources related to the composer, Mateo Flecha. Using these links the user can view a sample image or can navigate to the resources related to the composer Mateo Flecha.

The image shows a screenshot of a web browser window displaying the Cervantes Musica website. The website has a header with the logo "Cervantes Música" and navigation links: Search, Help, About. A sidebar on the left lists categories: Categories, Instrument, Song, Dance, Composer, Bibliography, Musical Reception, Virtual Tour, and Log Out. The main content area is titled "Mira Nero de Tarpeya" and includes tabs for Explanation, Music Note, Audio, and Cervantes Text. Below these tabs, there is a link to "Click here to add a new Music Note" and a section titled "Or Please select a Music Note entry to" with three options: Mira Nero, de Tarpeya, Mira Nero, de Tarpeya2, and Mira Nero, de Tarpeya3.

Overlaid on the website is a window titled "Finale NotePad 2005a". The window shows a music score for the song "Mira Nero, de Tarpeya" (Cancionero musical de Lope de Vega Vol.III) by Mateo Fl. The score is for three voices: Tiple, Alto, and Tenor. The Tiple part starts with a treble clef and a key signature of one flat. The Alto and Tenor parts start with a bass clef and a key signature of one flat. The lyrics are: "Mi ra Ne ro de Tar pe ya" for Tiple and Alto, and "Mi ra Ne ro de Tar pe ya a Ro mia" for Tenor. The score is displayed on a staff with a 3/4 time signature.

Figure 6: Music score for the song *Mira Nero de Tarpeya*



**Figure 7: Related links for the composer Mateo Flecha**

The automatically created links can help the user navigate from the song to the resources related to its composer. The user can access composer's images, texts about his life, and list of his works. Again if the library has any information related to these works then interlinks will be automatically created. The Figure 8 shows the interlinks that link the composer to his works. Thus the interlinkages logically link the distinct resources, creating multiple access points, thereby helping the user navigate and discover related resources easily.

The screenshot shows a web browser window titled "Cervantes Musica - Mozilla Firefox". The address bar displays a local URL: `http://localhost:8080/CervantesMusic/listTopicSubitem.do?selectedItemId=9&selectedTopicID=3`. The website header features the "Cervantes Música" logo and navigation links for "Search", "Help", and "About".

On the left, a vertical sidebar contains a "Categories" menu with options: Instrument, Song, Dance, Composer, Bibliography, Musical Reception, Virtual Tour, and Log Out. The "Composer" category is selected.

The main content area is titled "Mateo Flecha" and includes sub-tabs for "Life", "Iconography", "Work", and "Bibliography". The "Work" tab is active, displaying a "Click here to add a new Work entry" link and a "List of Work follows" section. The first item in the list is "Mira Nero de Tarpeya".

A "Related Links" pop-up menu is overlaid on the "Mira Nero de Tarpeya" link, providing additional navigation options: "Song: Mira Nero de Tarpeya (1055)", "Introduction Text", "Music Notes", "Sample Music Note", "Bibliography", "Composers", and "Cervantes Text".

At the bottom of the page, a copyright notice reads: "Copyright © Center for the Study of Digital Libraries. All Rights Reserved".

**Figure 8: Related links for the song links the composer to the songs that he has composed**



## 6. SYSTEM DESIGN

The system was developed using user-driven iterative process [18] [8]. Initially HTML prototypes were used to get a better understanding of the requirements. The prototypes also helped the users and the stakeholders to visualize how the system would look and work, long before the system was actually built. Once the initial requirements were obtained, the system was built iteratively in small incremental phases. At the end of each phase, feedback was collected. Comments and feedback were incorporated into the system in the subsequent phases. Building in incremental phases allowed the stakeholders to get a feel for the system while it was being developed and hence any change in the requirements were identified as early as possible, reducing the amount of rework required.

### 6.1. Technology Components

The following components are used in the implementation of the digital library system:

1. JSP and Servlets are used to build the user interface for the system.
2. Tomcat is used as the servlet and JSP container and is responsible for executing the servlets and the JSPs, generating dynamic web pages [3].
3. The Struts framework is used to implement the model-view-controller architecture [2].
4. A MySQL database and the file system are used as the storage repository [25].
5. Lucene is used as the information retrieval library and is used to index the texts [21].
6. AJAX tags are used implement the asynchronous request and response functionality for the system [1].

7. Walden's Paths is used to create virtual paths [16].

## **6.2. System Objectives**

The objectives of the system are:

1. The system will allow authorized users to upload new artifacts through web forms accessible over the Internet. The system will support uploading of different types of artifacts like texts, image, audio files, music scores, etc.
2. The system will be able to handle European characters and will use UTF-8 encoding. The system will preserve these European characters, while storing the texts. Additionally, it will represent the European characters correctly while displaying the texts.
3. The system will store and organize the uploaded artifacts using a MySQL database and the file system.
4. The system will use a MySQL database to store the metadata.
5. The system will provide access to the added artifacts by creating a browsing index and presenting the information in a structured manner.
6. The system will generate links (called "interlinks") to the related resources using the metadata repository. The related links will be shown as a tooltip, on demand, using AJAX technology. The system will also support the identification and linking of synonyms and multi-word phrases.
7. The interlinks will be displayed in an unobtrusive manner.
8. The display of the interlinks will have minimum impact on the response time.
9. The library will be able to maintain and create the interlinks while allowing for progressive addition of new information.

### 6.3. Organization of the Digital Library

The artifacts are organized under hierarchical groups like category, item and topic to facilitate easy information discovery and browsing. At the highest level all the artifacts are grouped under eight categories. The categories supported by the digital library are:

1. Instruments: artifacts and information pertaining to different musical instruments that have been referred to by Cervantes in his works.
2. Songs: information regarding the different songs that have influenced Cervantes.
3. Dances: resources related to the dances that have been referred to in Cervantes' texts.
4. Composers: the composers who have influenced Cervantes and his work.
5. Bibliography: bibliographical entries related to instruments, songs, and dances that have been referred to in Cervantes texts.
6. Musical Reception: bibliographical entries that have been influenced by Cervantes or refer to his works.
7. Cervantes Text: Cervantes' works are stored under this category.
8. Virtual Tour: links to virtual paths, constructed and hosted using Walden's Paths. This allows the information to be grouped and presented in different manner, catering to the interests of diverse scholars, thus opening up the digital library to unique interpretive perspectives.

The artifacts under a category are further grouped as items. An item defines a unique logical entity and may represent an instrument, a song, a dance, or a composer. The item is identified by its name. For example the category "Instruments" contains items like Arpa, Guitar, etc. The artifacts under each item are stored under different topics like image, audio, text, etc. Thus the actual artifacts are grouped under topics which in turn

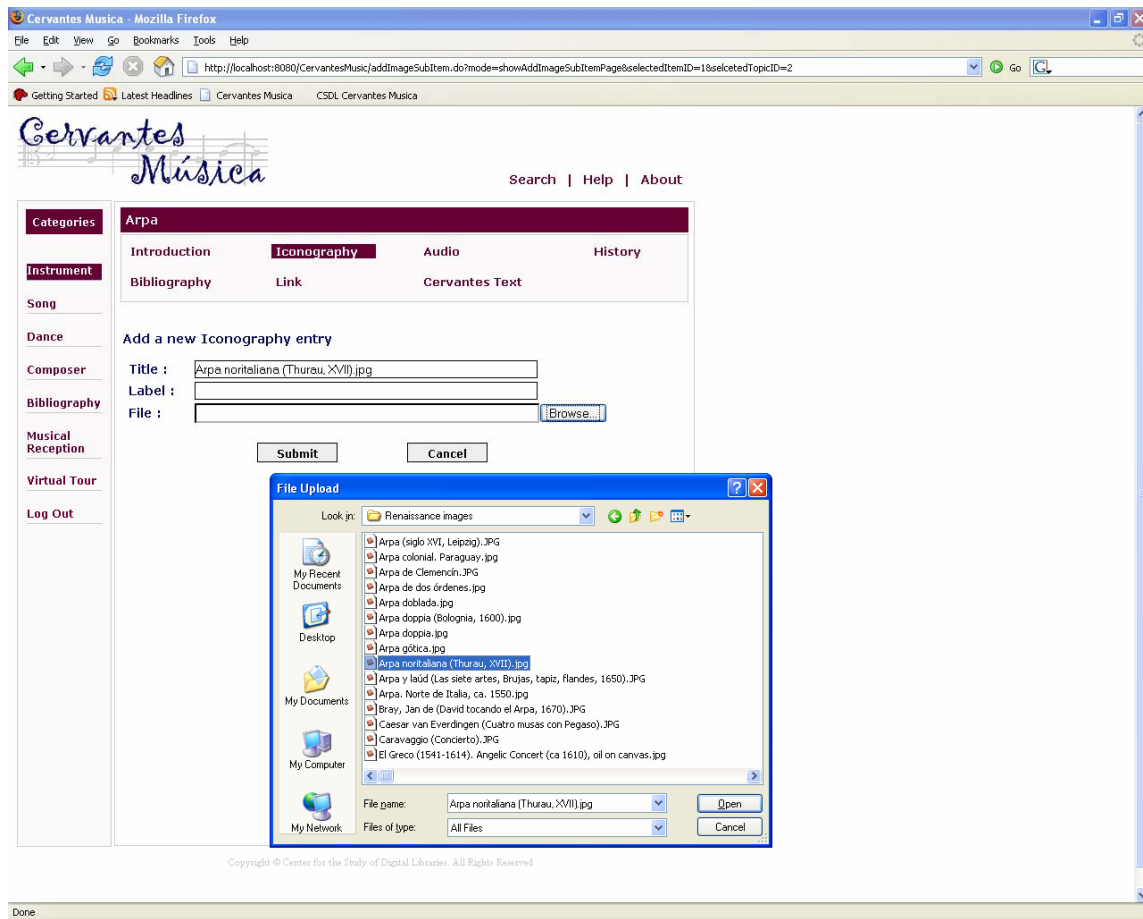
are organized under items, and these items are finally grouped by categories. The topics under an item depend on the category to which the item belongs to. For example an item under category “Instruments” will have topics like introduction, audio, image, text, and bibliography but an item under the category “Composer” will have topics like life, image, work, and bibliography. Each item added to the digital library is identified by a unique item identifier. Additionally, each artifact added under an item is assigned a sub-item identifier that is unique among all the artifacts under that item. Thus all the artifacts including texts, audio files, images, musical scores, etc., are uniquely identified by the combination of item identifier and sub-item identifier and are referred to as sub-items.

Figure 1 shows the various categories and items under the instrument category. The categories are listed on the left menu. When a category is selected, instruments in this case, all the items present under that category are displayed. When an item is selected, the system displays the various topics that are available for the item. Then, when one of the topics is selected, the system lists the artifacts present under that topic. Figure 2 shows the various topics available under the “Instruments” category and the sub-items present under topic “Iconography” for the instrument Arpa.

#### **6.4. Implementation of the Digital Library**

The digital library is developed using 3 tier model and is organized into the following three layers: the user interface layer, the business logic layer and the storage layer.

The user interface layer provides the interface for users to browse and add information. JSP and servlets are used to create the web interface. Java scripts are used to support client side processing. Web forms are used to support addition of information into the digital library. Figure 9 shows the web form for uploading artifacts.

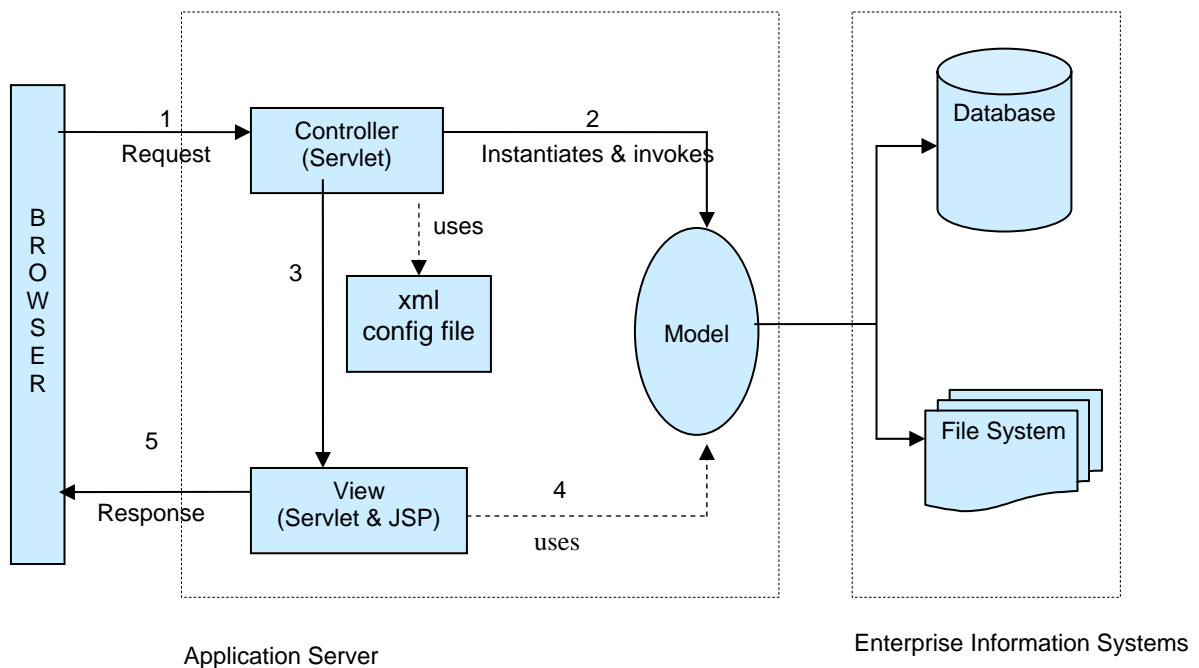


**Figure 9: User interface for uploading artifacts**

The business logic layer deals with the management of the artifact and the work flow. It handles the addition of artifacts and ensures that they are properly stored along with their associated metadata. It creates the browsing index and provides access to the material available in the digital library. This layer also handles indexing of the texts, logging, and user authentication. It is also responsible for the creating the interlinking and generating the related links.

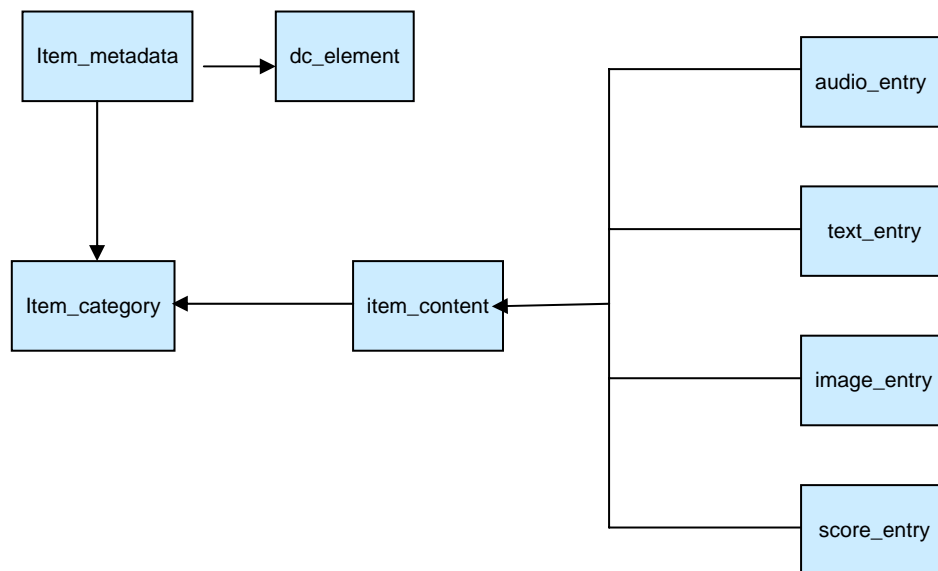
The system design is based on the Model View Controller (MVC) architecture. The model represents for the business components with its associated business logic and data representation. The view represents the user interface and consists of the JSP and the

servlets. The controller is responsible for the control flow. The Struts framework is used to implement this MVC pattern. As shown in Figure 10, the client request is handled by the controller servlet. It instantiates and invokes the business component, referred to as models, to handle the user requests. Finally the controller forwards the request to the appropriate view. The view retrieves the data generated by the models and displays it to the user. The view contains the logic to interpret the data and display it appropriately. But it is not responsible for any business processing. This ensures a clean separation of concern between view, model and the controller. This makes the system more flexible and reduces the impact of change in one component on the others.



**Figure 10: Model view controller architecture**

The storage layer is responsible for storing the artifacts, texts and their associated meta data. The storage layer consists of a MySQL database and the file system. All the uploaded artifacts, except the texts, are stored in the file system. Their associated metadata are stored in the MySQL database. The texts are also stored in the database. Figure 11 represents the data model for the metadata repository.



**Figure 11: Data model for the metadata repository**

For each item added to the digital library, an entry is added to the `item_category` table, consisting of a unique item identifier and the category identifier. The item level metadata is stored in the `item_metadata` table, using the metadata identifier defined in the `dc_element` table. For each artifact added to the system, an entry is added to the `item_content` table consisting of the item identifier, representing the item to which the artifact belongs, and a sub-item identifier that is unique among all the artifacts under the

item. Since the artifacts or the sub-items are of diverse media types, some of the metadata attributes associated with them will be specific to the media types, while other attributes are common across all the media types. The metadata attributes common across all these media types are stored in the `item_content` table. But the type specific metadata attributes are stored in the media specific tables. For example metadata like user added, date added, etc., can be stored in the `item_content` table. However, sub-items like text may have type specific metadata like author, book title, and chapter. These are stored in the corresponding type table; `text_entry` in this case. Some of the type specific tables used by the system are `text_entry`, `audio_entry`, `image_entry`, `score_entry`, etc. Currently the system enforces limited metadata. But it supports the addition and display of additional metadata by providing a free-form text box for each sub item.

The business components use the file manager and database manager to interface with the storage layer. The file manager and database manager implement the logic required for interacting with the storage layer. This ensures separation of concerns and that this logic is not duplicated among business components. The database manager is responsible for establishing the connection to the database, executing the insert, update and select statements on the database, and managing the transactions. Since establishing a connection to the database is an expensive operation, connection pooling is used. When the system starts a configurable number of connections to the database are created and maintained as a connection pool. The database manager retrieves a connection from the pool when required and returns the connection back to the pool when done. The file manager is used by the business components to save the uploaded artifacts in the file system. As the file names may have arbitrary characters that may not be compatible across different operating systems, the file manager modifies the filenames before saving the files to the file system. Since each file is uniquely identified by the combination of item id and sub-item id, the original name is replaced by this combination of the item id followed by the sub-item id. While the filenames are modified, the file extensions are preserved. The file manager creates a new directory for each item, named using the item id, and stores all the attributes for that item under this directory.



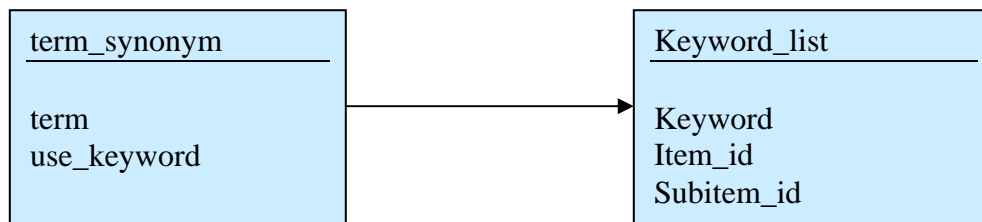
The system supports European-language characters by ensuring that whenever the text is transmitted as bytes it is encoded using UTF-8 format. When a text is added using the web forms, the browser is instructed to encode the text using UTF-8. When the request is received at the server, the server is instructed to decode the submitted text using UTF-8, thus ensuring that the European-language characters are properly recognized. Since this has to be done for all the requests, an intercepting filter is used that intercepts all the requests and ensures that they are decoded using UTF-8. Once done, it forwards the request to the appropriate business components for further processing. When the texts are stored in the database, the database is instructed to store the text using UTF-8 encoding. When the texts are retrieved from the database, it is again instructed to use UTF-8 to decode the texts. Finally, when the texts are displayed on the browser, it is ensured that the server encodes the texts using UTF-8 and the browser decodes them using UTF-8. Since all the encodings and decodings are done using UTF-8, this guarantees that all the characters including the European characters are properly identified, stored and displayed.

## **6.5. Implementation of Interlinking**

The entire process of creating interlinks and presenting the related links can be broadly classified into four major steps. The first is maintaining the list of item names for which information exists in the digital library. The second is a batch job, which identifies the reference of these terms in all the texts present in the digital library. The third step is a run time process, which, while displaying a text, embeds the terms that need to be linked with a dummy hyperlink (i.e., hyperlink without any specific target). This step uses the data from the batch job to identify the terms that should be presented with the hyperlink for any text. The final step generates the actual related links for a term and is invoked only when the user clicks on any of the above dummy hyperlinks. A detailed description of these steps follows.

### 6.5.1. Maintaining the Keyword List

For the system to provide related links it should be able to identify the terms for which information exists in the digital library. This is achieved by maintaining a keyword list. To identify the variation in names a synonym list is also maintained. The system depends on the user to provide a list of synonyms for the item being added. This may include alternate names for the item or just variations in the spelling of the item name.

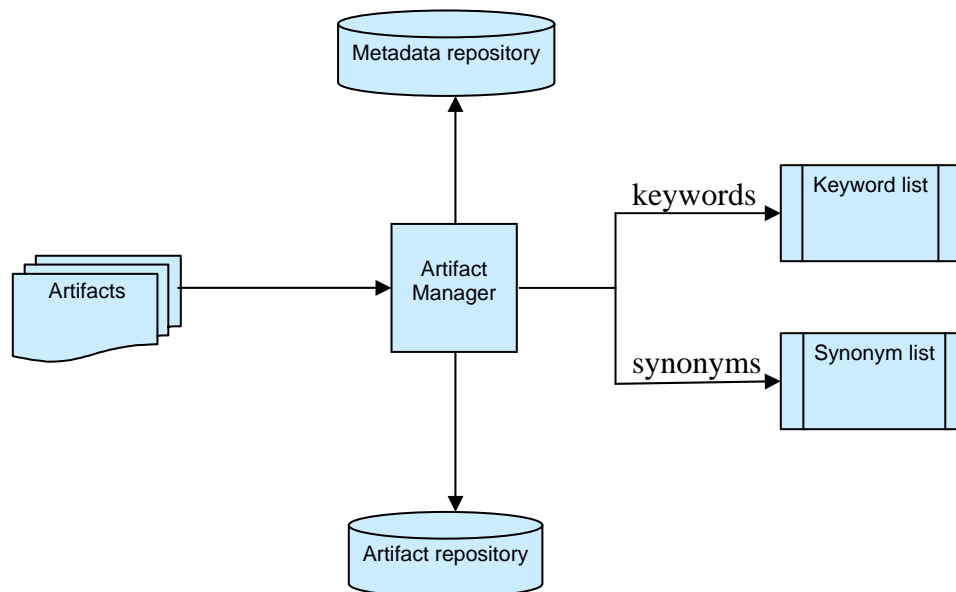


**Figure 12: The keyword list data model**

Figure 12 shows the data model for the keyword list and the synonym list. When a new item is added to the digital library its name or title is added to the keyword\_list table. For example if an instrument is added to the digital library, the name of the instrument is added to this keyword\_list table. For each entry, its corresponding item id and sub item id are also stored. If it represents a sub-item both its item id and sub-item id are populated. And if it refers to an item then only its item id is populated. Its synonyms or the variations in its name are stored in the term\_synonym table. The term column in term\_synonym refers to the variations and the use\_keyword column links this variation to the actual keyword in the keyword\_list table. The term\_synonym table stores all the variations including the actual keyword. Thus for any term in the term\_synonym table, the use\_keyword value can be used to link it to the keyword\_list table and hence to its

corresponding item id and sub-item id. A schematic diagram of this process is shown in Figure 13.

Thus term\_synonym table contain all the terms for which information exists in the digital library. Any reference to these terms in the texts need to be identified and hyperlinked to the related resources. In the following sections the terms in both the term\_synonym and keyword\_list will be referred to as keywords.



**Figure 13: Schematic diagram for maintaining keyword list and synonym list**

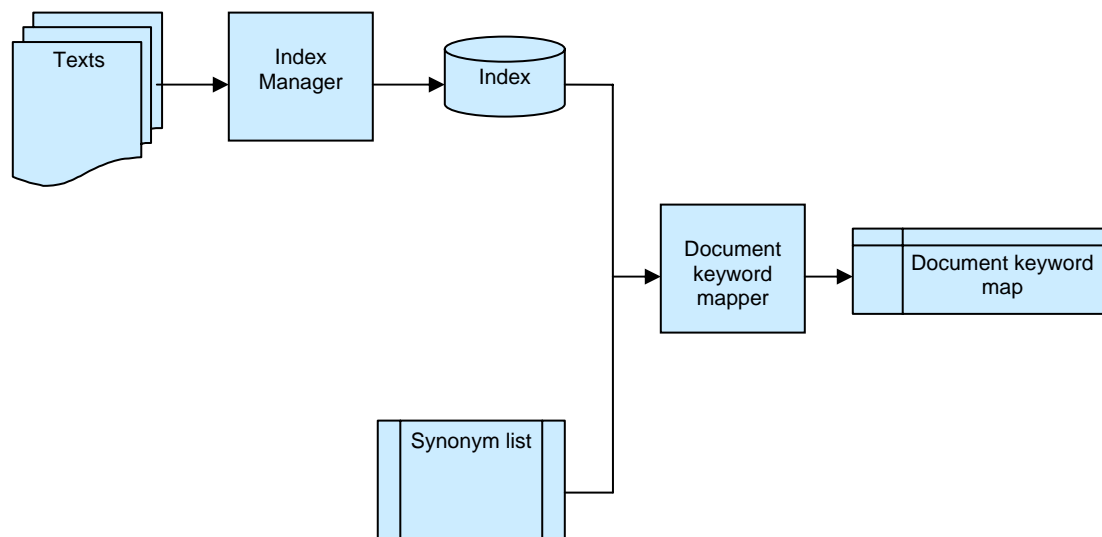
### 6.5.2. *Document Keyword Mapping Batch Job*

The document keyword mapping is created by indexing all the texts and finding the references of each term in the keyword list among all the texts. This is done offline using a batch process.

All the texts added to the digital library are stored in the `text_entry` table and each text is uniquely identified by a text identifier. The batch job retrieves the texts that need to be indexed, from the database. It also retrieves their corresponding text identifiers. The `text_entry` table has a flag to identify the texts that are new and have not been indexed. Only these new texts are retrieved and indexed, to support incremental indexing. Lucene is used for indexing the texts. While indexing the texts, the text identifiers are also indexed as keywords. This enables the retrieval of associated text identifiers for the matching texts, during a search. While creating indexes, Lucene may create multiple index files. Multiple index files increase the search time. Hence at the end of indexing, the Lucene index is optimized by merging all the index files into a single file. This optimization improves the search performance. To support large amount of texts, the Lucene index is stored in the file system instead of using the in-memory index. Some of the reasons for using Lucene as the information retrieval library are: it is open source software, is highly scalable and is implemented in Java. Since it is implemented in Java, it supports multiplatform deployment. Lucene also provides a simple interface for indexing and searching and supports incremental indexing.

The index manager provides the interface necessary to index a text and search for a term. All the logic related to setting up and using the Lucene APIs is implemented in the index manager. The components that require index and search functionality can use the simple interface provided by index manager rather than worrying about setting up and using the Lucene API. Since all the components access the Lucene API through the index manager and not directly, if need arises Lucene can be replaced with another information retrieval library without affecting the components.

Once the indexing is done, the keyword\_doc\_mapping table is populated. The batch job retrieves all the keywords from the term\_synonym table. For each keyword, the Lucene search API is invoked, which returns the matching texts. Since text identifiers were also indexed along with the texts, these associated text identifiers are also retrieved. These text ids uniquely identify the texts that have references to the searched keyword. This information is recorded in the keyword\_doc\_mapping table. Figure 14 gives a schematic overview of this process. A sample of the keyword\_doc\_mapping table is shown in Figure 15.



**Figure 14: Generation of document keyword map**

term	text_id
arpa	15
arpa	12
arpa	23
arpa	20
arpa	8
arpa	11
arpa	10
arpa	7
arpa	13
arpa	25
arpa	14
guitarra	13
guitarra	14

**Figure 15: A sample keyword document mapping table**

#### 6.5.3. Runtime Display of Texts with Dummy Hyperlinks

While displaying a text the system needs to identify the keywords from the keyword list that is present in the text. This can be done by matching all the keywords in the keyword list against the text. When the number of keywords is large, this can be very expensive and may affect the response time. Thus keyword\_document\_mapping table is used instead. Using the text identifier of the text to be displayed, the list of keywords present in the text can be obtained directly from the keyword\_document\_mapping table.

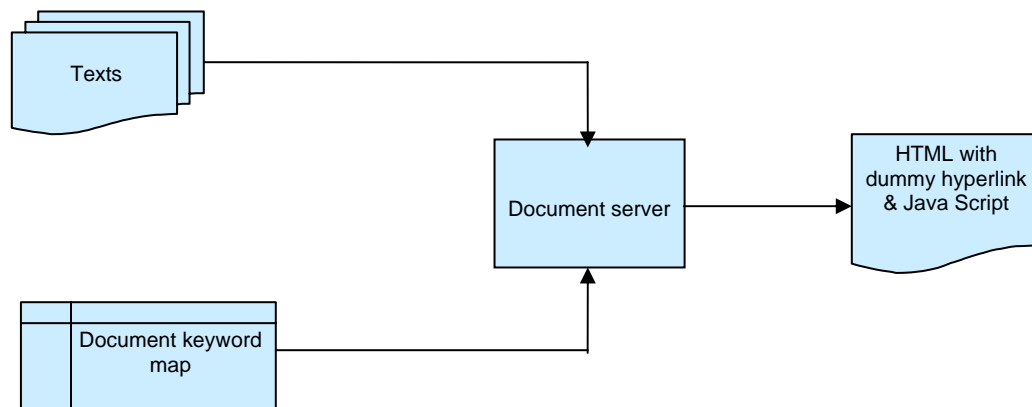
Once the list of keyword present in the text is known, their occurrences in the text are identified and are provided with a dummy hyperlink. A regular expression is built using these keywords and the JDK regular expression engine is used to replace these keywords by a dummy hyperlink of the following format:

```
<a href="javascript:nop" class="cerhyperlink"> keyword </a>
```

where the keyword stands for the keyword that is being hyperlinked.

This hyperlink does not link to any particular target, instead when the user clicks on it, the event is intercepted by client side java script which does the necessary processing to obtain and display the related links using a tooltip.

These keywords are shown in contrasting color to differentiate them from the rest of the text. They are also shown with an underline to make it apparent that they are hyperlinks and are clickable. Figure 16 represents a schematic overview of this process.



**Figure 16: Display of texts with dummy hyperlinks**

#### 6.5.4. *Display of Related Links Tooltip*

As described in the previous section the system displays the text with the keywords identified and hyperlinked. But the related links are generated and displayed only when the user clicks on any of these keywords.

When the user clicks on any of these hyperlinked keywords the event is intercepted by client side java script function. The java script function parses the hyperlink statement

and retrieves the actual keyword. AJAX technology is used to send this keyword to the server and to receive the related links asynchronously in the background without page reload. The post method is used to avoid URL encoding problems. Since the keywords may contain European characters, request encoding is set to UTF-8 to preserve these European characters.

When the request is received at the server, the keyword is retrieved from the request parameters. HyperlinkBuilder component is used to build the links to the related resources. HyperlinkBuilder uses the keyword passed as an argument, to find the corresponding use\_keyword entry from the synonym\_list table. This in turn is used to find the corresponding item\_id from the keyword\_list table. This item identifier represents the item whose name or one of its synonyms is the keyword. The item\_content table stores the structural metadata of all the sub items and hence can be used to identify all the sub items which belong to this item. Using these sub items the distinct list of topics to which they belong is identified and links to these topics is generated. For example if the item has some related resources which are image then link to view the images is added to the related link list. Further the format of these sub items is also noted. If they are of formats like image, audio etc then link to a sample image or audio is also added to the related link list. This sample audio or image window is shown in a new page right on top of the text. This allows the user to view a sample image or listen to a sample audio clip without leaving the text. Once the complete list of related topics is generated the AJAX response is sent back to the client. The response is sent using the “text/xml” mime type and the encoding is set to UTF-8. A sample of this response is shown in Figure 17.

```
<?xml version="1.0" encoding="UTF-8" ?>
<ajax-response>
<response>
<item>
<name>
<![CDATA[callout Header]]></name><value>
<![CDATA[ <b>
Instrument: &nbsp; <a href='displayItem.do?selectedItemID=1'>Arpa</a>
```



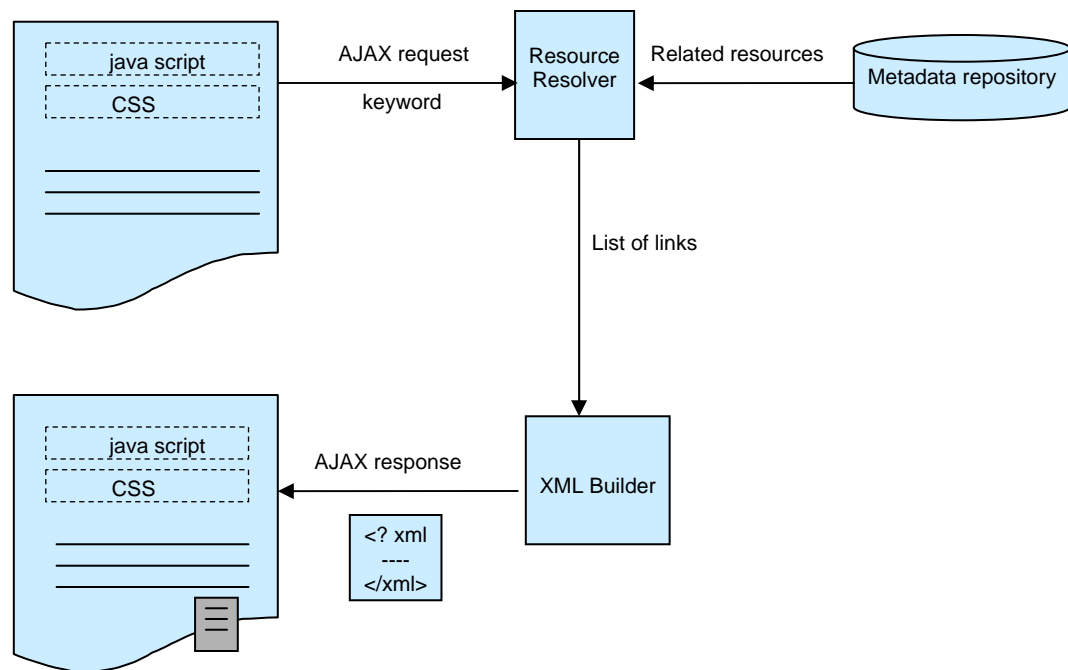
```

<br><hr>
<a href='listTopicSubitem.do?selectedItemID=1&selcetedTopicID=1'>
  Introduction Text</a>
<br> <hr>
<a href = 'listTopicSubitem.do?selectedItemID=1&selcetedTopicID=2'>
Iconography</a>
<br> <hr>
<a href="javascript:openWindow('resources/1/11.png');"> Sample Image</a>
<br> <hr>
<a
href='listTopicSubitem.do?selectedItemID=1&selcetedTopicID=3'>Audio</a>
<br> <hr>
<a href="javascript:openWindow('resources/1/111.mp3');"> Sample Audio</a>
<br> <hr>
<a href='listTopicSubitem.do?selectedItemID=1&selcetedTopicID=6'>External
Links</a> </b>]]>
</value></item>
</response>
</ajax-response>

```

**Figure 17: A sample AJAX response**

This response is received by the client as an xml document object. The xml document object is parsed using java script to obtain the related links. These related links are displayed in a tooltip just below the keyword clicked by the user. Cascading style sheets are used to control the look and feel of the tooltip. Figure 18 gives an overview of this process of generating related links. Displaying the related links on the same page, using a tooltip just below the keyword ensures continuity of action and minimum disruption for the users. The users can see the related topics without leaving the texts. They can also listen to a sample audio and view a sample image. Only if they are really interested in a topic, they can view all the resources for that topic, by clicking on the corresponding link in the tooltip.



**Figure 18: Generation and display of related links**

## 7. DISCUSSION AND FUTURE WORK

The system performed according to expectations and was able to identify the keywords and generate the links to the related resources correctly. Displaying the related links in a tooltip using AJAX technology allowed the user to view these links without leaving the text, thus ensuring minimum disruption and continuity of action. Displaying related links on a tooltip allowed the system to link a keyword to several related resources rather than a single target.

Distributing the entire process of creation of related links across batch, runtime and on demand phases seems to be working perfectly. Generating the document keyword mapping in a batch process and using this mapping to identify and provide hyperlinks while displaying the texts, ensured that the runtime overhead is minimized. The decision to generate the related links on demand also seems correct. The related links are generated only when the user explicitly expresses his interest by clicking on the keywords. Thus the only additional activities required while displaying the text is to identify the keywords, provide them with a contrasting color, underline them, and embed them using dummy hyperlinks. The use of regular expression to provide the dummy hyperlinks, for the identified keywords, also helped to minimize the runtime overhead. This mechanism allowed the system to display very large texts, with keywords identified and hyperlinked, with no perceivable delay to the user. The use of contrasting color and underlining of the keywords helped the users to identify the keyword easily. Generating related links only when the user clicks on the keyword ensured that there is minimum information overload for the user.

Since the interlinking is generated during runtime, no modification of the actual texts is required. The texts are stored in their original state, and the links are embedded at the runtime. This scales well with rapidly changing collection where some new links must be added and some old links removed from the texts. And since the related links are generated at the run time they represent the actual state of the collection at that time.

One of the limitations of using this mechanism to provide interlinking is that it cannot identify those items which are added after the last run of the batch process. The batch process generates the keyword document mapping. The batch process will only identify the references of the items that are present at the time of its execution. Thus the items added after the batch process will not be included in this keyword document mapping. While displaying a text, this keyword document mapping is used to identify the keywords present in that document and to hyperlink them. The items added after the batch process, will not be present in the mapping and hence will not be identified and interlinked. But this limitation can be minimized by running the batch process at regular intervals. The batch process can be scheduled to run automatically every night and update the document keyword mapping.

Another limitation of this mechanism is it can only provide links to related resources for texts displayed as HTML pages. It cannot embed the hyperlinks in text documents like Microsoft word or Acrobat pdf files.

The system supports identifying synonyms and variations in names and linking them to the related resources. But it depends on the user to provide a list of synonyms and alternate names. As a part of future work, suitable reference materials can be identified and algorithms developed to automatically extract the synonym list from them. Algorithms to identify and resolve variations in spelling can also be used to identify keywords spelled differently and link them to related resources.

Mechanisms can also be developed to control the type of links generated and displayed to the user. Generating links at runtime provides sufficient flexibility to control the type of links displayed, based on different parameters. For example user preference can be captured and used to display particular type of links while suppressing the others. The context in which the keyword is used adds significant meaning to the keyword. Context analysis techniques can be used to identify this context, which can help in generating links to additional related resources.

This technique can also be used to integrate and provide interlinking across multiple digital libraries. Authority list consisting of keywords and their authoritative source of information can be imported from other digital libraries. The terms on the authority list can be integrated with the keyword list, so that the reference to those terms can be identified. The resource resolver can use the authority list to provide the corresponding links. In a fully distributed environment, the resource resolver can pass these terms as arguments to the respective digital libraries and request for related links. It can then display these related links, provided by the other digital libraries, to the user. This will help in creating a fully integrated environment across digital libraries not just within a single digital library.

The digital library currently supports a very simple work flow. Artifacts uploaded are immediately archived and made available to users. A review phase can be added so that artifacts are made available only after they are approved by the reviewers. The authentication and authorization process can be improved by providing finer control on the permissions granted to users. More elaborate functionality to manage the artifacts present in the library can be incorporated.

## 8. CONCLUSION

In this thesis we have described a mechanism for interlinking related diverse media in a digital library. We have implemented a music digital library system. The library allows the authorized users to add various artifacts remotely using web forms. It builds a browsing index and make these artifacts accessible over the web in a well organized and structured manner. It automatically provides interlinking between related resources. This will help the humanities scholar in their research of the influence of music on Cervantes and his influence on subsequent writers and musicians.

We have implemented an interlinking mechanism that spans across batch, online and on-demand phases. Since the task of generating the related links is resource and time intensive, distributing the whole process across batch and online phases significantly reduces the runtime overhead and improves the response time. This mechanism allows the system to display very large texts, with keywords identified and hyperlinked, with no perceivable delay to the user. The metadata is preserved and used in generating related links to diverse media like images, audio files, music scores, texts, etc. The related links are displayed on demand using AJAX technology. This allows the user to view these links without leaving the text, thus ensuring minimum disruption and continuity of action.

We also have developed a generic interlinking framework which abstracts the domain independent logic for generating and displaying related links. This generic interlinking framework can be used by domain specific digital libraries to support interlinking of related resources.

## REFERENCES

- [1] "Ajax Tags," <http://ajaxtags.no-ip.info/>, Accessed 2nd Jan 2006.
- [2] "Apache Struts Project," The Apache Software Foundation, Forest Hill, MD, <http://struts.apache.org/>, Accessed 2nd Jan 2006.
- [3] "Apache Tomcat Documentation," The Apache Software Foundation, Forest Hill, MD, <http://tomcat.apache.org/>, Accessed 2nd Jan 2006.
- [4] N. Audenart, R. Furuta, E. Urbina, J. Deng, C. Monroy, R. Saenz, and D. Caaareaga, "Integrating Diverse Research in a Digital Library Focused on a Single Author," *Proc. of the 9th European Conf. on Research and Advanced Technology for Digital Libraries*, Vienna, Austria, 2005.
- [5] N. Audenart, R. Furuta, E. Urbina, J. Deng, C. Monroy, R. Saenz, and D. Careaga, "Integrating Collections at the Cervantes Project," *Proc. of the 5th ACM/IEEE-CS Joint Conf. on Digital Libraries*, Denver, CO, 2005.
- [6] D. Bainbridge, "The Role of Music IR in the New Zealand Digital Library project," *Proc. of the International Symposium on Music Information Retrieval*, Plymouth, MA, 2000.
- [7] D. Bainbridge, C. G. Nevill-Manning, I. H. Witten, L. A. Smith, and R. J. McNab, "Towards a Digital Library of Popular Music," *Proc. of the Fourth ACM Conf. on Digital Libraries*, pp. 161-169, Berkeley, CA, 1999.
- [8] S. Bødker, K. Grønbæk, and M. Kyng, "Cooperative Design: Techniques and Experiences from the Scandinavian Scene," in *Human-computer Interaction: Toward the Year 2000*. San Francisco, CA: Morgan Kaufmann Publishers Inc, 1995, pp. 215-224.
- [9] X. Chen, D. K. Kim, N. Nnadi, H. Shah, P. Shrivastava, M. Bieber, Il Im, and Yi-Fang Wu, "Digital Library Service Integration," *Proc. of the 3rd ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 384-384, Houston, TX, 2003.
- [10] G. Crane, E. David, A. Smith, and C. E. Wulfman, "Building a Hypertextual Digital Library in the Humanities: A Case Study on London," *Proc. of the 1st ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 426-434, Roanoke, VA, 2001.
- [11] S. J. Cunningham, N. Reeves, and M. Britland, "An Ethnographic Study of Music Information Seeking: Implications for the Design of a Music Digital Library," *Proc. of the 3rd ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 5-16, Houston, TX, 2003.

- [12] "DSpace Federation," <http://www.dspace.org/>, Accessed on 29 Nov 2005.
- [13] J. W. Dunn and E. J. Isaacson, "Indiana University Digital Music Library Project," *Proc. of the 1st ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 452, Roanoke, VA, 2001.
- [14] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-Specific Keyphrase Extraction," *Proc. of the Sixteenth International Joint Conf. on Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers, 1999.
- [15] R. Furuta, S. S. Kalasapur, R. Kochumman, E. Urbina, and R. Vivancos-Perez, "The Cervantes Project: Steps to a Customizable and Interlinked On-Line Electronic Variorum Edition Supporting Scholarship," *Proc. of the 5th European Conf. on Research and Advanced Technology for Digital Libraries*, Darmstadt, Germany, 2001.
- [16] R. Furuta, F.M. Shipman III, C.C. Marshall, D. Brenner, and H. Hsieh, "Hypertext Paths and the World-Wide Web: Experiences with Walden's Paths," *Proc. of the Eighth ACM Conf. on Hypertext (HT '97)*, pp. 167-176, Southampton, United Kingdom, April 1997.
- [17] G. Geisler, S. Giersch, D. McArthur, and M. McClelland, "Creating Virtual Collections in Digital Libraries: Benefits and Implementation Issues", *Proc. of the 2nd ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 210-218, Portland, OR, 2002.
- [18] J. Grudin, "Interactive Systems: Bridging the Gaps between Developers and Users," *Computer*, vol. 24, no. 4, pp. 59-69, April 1991.
- [19] S. Jones and G. Paynter, "Topic Based Browsing Within a Digital Library Using Keyphrases," *Proc. of the Fourth ACM Conf. on Digital Libraries*, pp. 114-121, Berkeley, CA, 1999.
- [20] C. Lagoze, "Core Services in the Architecture of the National Science Digital Library," *Proc. of the 2nd ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 201-209, Portland, OR, 2002.
- [21] "Lucene Overview," The Apache Software Foundation, MD, USA, <http://lucene.apache.org/java/docs/>, Accessed 2nd Jan 2006.
- [22] R. J. McNab, L. A. Smith, D. Bainbridge, and I H. Witten, "The New Zealand Digital Library Melody Index," *D-Lib Magazine*, May 1997.



- [23] W. H. Mischo, T. G. Habing, and T. W. Cole, "Integration of Simultaneous Searching and Reference Linking Across Bibliographic Resources on the Web," *Proc. of the 2nd ACM/IEEE-CS Joint Conf. on Digital Libraries*, pp. 119-125, Portland, OR, 2002.
- [24] C. Monroy, R. Kochumman, R. Furuta, and E. Urbina, "Interactive Timeline Viewer: A Tool to Visualize Variants," *Proc. of 2nd Intl' Workshop on Visual Interfaces to Digital Libraries (in conj. with JCDL '02)*, Portland, OR, July 2002.
- [25] "MySQL Documentation," <http://www.mysql.com/>, Accessed 2nd Jan 2006.
- [26] "New Zealand Digital Library," <http://www.nzdl.org> , Accessed 29th Nov 2005.
- [27] J. J Pastor, "Musica y literatura: la senda retorica. Hacia una nueva consideracion de la musica en Cervantes," Ph. D dissertation, Universidad de Castilla-La Mancha, Toledo, Spain, 2005.
- [28] M. S. Patton and D. M. Mimno, "Services for a Customizable Authority Linking Environment," *Proc. of the 4th ACM/IEEE-CS joint conf. on Digital Libraries*, pp. 420-420, Tuscon, AZ, 2004.
- [29] L.D. Paulson, "Building Rich Web Applications with Ajax," *Computer*, vol. 38, issue 10, pp. 14-17, Oct 2005.
- [30] B. Pritchett, "KeyLinking: Dynamic Hypertext in a Digital Library," *Proc. of the Fifth ACM Conf. on Digital Libraries*, pp. 242-243, San Antonio, TX, 2000.
- [31] G. Salton, *Automatic Text Processing – The Transformation, Analysis and Retrieval of Information by the Computer*. Reading, MA: Addison-Wesley Publishing Co, 1989.
- [32] K. Silwa, *Documents Cervantinos: Nueva recopilacion; lista e indices*. New York: Peter Lang, 2000.
- [33] E. Urbina, Ed., "The Cervantes International Bibliography Online," Center for the Study of Digital Libraries, Texas A&M University, <http://csdl.tamu.edu/cervantes.>, Accessed on Nov 29 2005.
- [34] E. Urbina, Ed., "The Cervantes Project," Center for the Study of Digital Libraries, Texas A&M University, <http://csdl.tamu.edu/cervantes.>, Accessed on Nov 29 2005.
- [35] I. H. Witten, D. Bainbridge, and S. Boddie, "Greenstone: Open Source DL Software," *Communications of the ACM*, vol. 44, no. 5, pp. 47, 2001.

- [36] I. H. Witten, S. Boddie, D. Bainbridge, and R. J. McNab, “Greenstone : A Comprehensive Open-Source Digital Library Software System,” *Proc. of the Fifth ACM conf. on Digital Libraries*, pp. 113-121, San Antonio, TX, 2000
- [37] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*. San Francisco, CA: Morgan Kaufmann Publishing, 1999.
- [38] A. J. Wolpert, “The Future of Electronic Data,” *Nature*, vol. 420, pp. 17-18, 2002.

## VITA

Contact Details	Manas Sourava Singh	
Permanent Address	SF 11, Vanivihar, BBSR, Orissa, India 751004	
Email Address	manassourav@tamu.edu, manassourav@yahoo.com	
Education	M.S., Computer Science Texas A&M University (TAMU) College Station, TX	May 2005
	B.E., Civil Engineering R.E.C Rourkela, Rourkela, India	May 1998